

Absolvovanie individuálnej odbornej praxe

Individual Professional Practice in the Company

Zadání bakalářské práce

Student: **Patrik Veteška**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Absolvování individuální odborné praxe**
Individual Professional Practice in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: ABB s.r.o.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Peter Chovanec**

Konzultant bakalářské práce: Ing. Jan Mináč

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prehlasujem, že som túto bakalársku prácu vykonal samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal

V Ostrave 2014

.....
Vojta

Rád by som týmto poďakoval svojmu konzultantovi Ing. Jánovi Mináčovi, za prejavení dôveru a poskytnutie možnosti pracovania na bakalárskej práci vo firme ABB s.r.o. Ďalej by som chcel poďakovať svojmu vedúcemu bakalárskej práce Ing. Petrovi Chovancovi za užitočné podnety, rady a pripomienky pri písaní tejto práce. Takisto by som rád poďakoval všetkým svojím kolegom zo softwarového oddelenia za vrelé prijatie do kolektívu.

Abstrakt

Táto bakalárska práca sa zaoberá popisom mojej pracovnej činnosti vo firme ABB s.r.o., kde som túto prax vykonával. Behom práce som riešil rozličné úlohy v oblasti vývoja softwaru v technológii .NET, ktoré tu opíšem. Výsledkom môjho snaženia spoločne s ostatnými ľuďmi podieľajúcimi sa na vývoji je aplikácia nasadená na internom serveri firmy používaná všetkými zamestnancami.

Kľúčové slová: C#, .NET, Javascript, PetaPoco, Scrum

Abstract

This bachelor thesis deals with the summary of my working activities at a company ABB s.r.o., where I carried out this practice. During work I was dealing various tasks in a field of Software Development in a .NET framework, which I will describe here. The result of my effort, together with other colleagues involved in development, is an application, situated on internal server of the company and it is using by all of the employees.

Keywords: C#, .NET, Javascript, PetaPoco, Scrum

Zoznam použitých skratiek a symbolov

CZOPC	– Operačné centrum Česká republika
C#	– Programovací jazyk
.NET	– Software framework developed by Microsoft
CSS	– Cascading Style Sheets
PETAPOCO	– Objektovo-relačné mapovanie pre .NET a Mono
SCRUM	– Metóda riadenia agilného vývoja softwaru
PDF	– Portable Document Format

Obsah

1	Úvod	4
2	Popis firmy ABB	5
2.1	Popis odborného zamerania firmy	5
2.2	Popis pracovného zaradenia študenta	6
3	Použité technológie a prístupy	7
3.1	.NET	7
3.2	JavaScript	7
3.3	Scrum	7
4	Zoznam úloh zadáných študentovi počas odbornej praxi	9
4.1	Projekt "Monthly record application"	9
4.2	Vykonané úpravy, rozšírenia a ich stručný popis	10
4.3	Práca s aktivitami na priradených projektoch, kalkulácie hodín	10
4.4	Úprava vzhľadu a šírky hlavnej tabuľky pre projekty	11
4.5	Povolenie a blokovanie špeciálnych kláves	11
4.6	Označovanie hodnôt a posielanie informačných emailov	12
4.7	Centrovanie vyskakovacích okien	12
4.8	PetaPoco	13
4.9	Projekt mohol začať, alebo skončiť hocikedy v priebehu mesiaca	13
4.10	Dynamické posúvanie tabuľky s projektami	14
4.11	Refresh stránky	15
4.12	Prepísanie treeview do dropdownlistov a rozšírenie stromovej štruktúry	15
4.13	Posúvanie popupov spoločne so scroll barom a zošednutie obrazovky za popupmi	17
4.14	Export údajov do excelu obsahujúcich worksheets	17
4.15	Problém so špeciálnymi znakmi a zhodenie aplikácie na produkčnom serveri	19
5	Využitie znalosti a skúsenosti	20
5.1	Chýbajúce znalosti a skúsenosti	20
6	Záver	21
7	Literatúra	22

Zoznam obrázkov

1	Nové sídlo Divízie procesnej automatizácie v Ostrave	5
---	--	---

Zoznam výpisov zdrojového kódu

1	Blokovanie klávesnice Enter a zobrazovanie popupu	12
2	Blokovanie buniek pre vkladanie hodnôt	14
3	Dynamické posúvanie tabuľky	15
4	Refresh stránky	15
5	Zmena indexu v gridview	16
6	Vkladanie hodnôt do dropdownlistov a zoradenie	16
7	Export údajov do excelu	18
8	Posielanie špeciálnych znakov zo C# do javascriptu	19

1 Úvod

Vyvrcholením môjho trojročného štúdia je bakalárska práca, ktorú som vykonával vo firme ABB. Bakalársku prax som si zvolil aby som si overil svoje vedomosti, ktoré som nadobudol v škole a na vlastnej koži si vyskúšal prácu v tíme na komerčnom projekte. Taktiež bolo mojím cieľom rozšíriť si vedomosti a získať čo najviac praktických skúseností, ktoré by som v budúcnosti využil a ktorých nebolo málo.

V prvej časti budem opisovať pracovné a podnikateľské zameranie firmy, v ktorej som túto prax vykonával a neskôr opíšem moje pracovné zameranie v tejto firme.

Hlavná časť práce sa zaoberá popisom webovej aplikácie „Monthly Record Application“, ktorá je primárne vytvorená na zaznamenávanie odpracovaných hodín, ktoré strávili jednotliví zamestnanci pri pracovnej činnosti na zadaných projektoch. Výstupom tejto aplikácie sú mimo iného aj reporty v určitom rozsahu dní o zaznamenatej pracovnej činnosti a úlohách spojených s touto činnosťou od ktorej sa odvíjajú jednotlivé výplaty.

Záver je určený na popis praktických a teoretických skúseností, ktoré som v tejto firme získal a využíval v priebehu odbornej praxi. Ďalej tým, ktoré mi chýbali a ktoré bolo nutné ešte rozvinúť. Nakoniec popisujem moje dosiahnuté výsledky a celkové zhodnotenie prínosov tejto odbornej praxi.

2 Popis firmy ABB

ABB je predná svetová spoločnosť poskytujúca technológie pre energetiku a automatizáciu, ktoré umožňujú energetickým a priemyslovým podnikom zvyšovať výkonnosť pri súčasnom znížení dopadu ich činnosti na životné prostredie. V ČR pôsobí od roku 1970 a nachádza sa vo viac ako 100 krajinách.



Obr. 1: Nové sídlo Divízie procesnej automatizácie v Ostrave

2.1 Popis odborného zamerania firmy

Divízia Výrobky pre energetiku

Predstavuje kľúčové komponenty pre prenos a rozvod elektrickej energie (výroba a dodávky rozvodní a prístrojov vn, transformátorov, senzorov...)

Divízia Systémy pre energetiku

Poskytuje komplexné dodávky a projektovanie pre energetiku, systémy merania a regulácie v energetike, káblové systémy.

Divízia Automatizácia výroby a pohony

Výrobky, riešenia a s nimi súvisiace služby, ktoré zvyšujú priemyslovú produktivitu a energetickú účinnosť (zahrňuje motory, generátory, meniče frekvencie, usmerňovače, programovateľné automaty, roboty...).

Divízia Výrobky nízkeho napätia

Táto divízia vyrába a predáva prístroje a rozvádzače s napätím do 1000V, ktoré zaisťujú bezpečnosť osôb a ochranu inštalácií a elektrických zariadení pred preťažením.

Divízia Procesná automatizácia

Poskytuje najlepšie riešenia pre riadenie a optimalizáciu prevozu a aplikačné skúsenosti špecifické pre priemyslové odvetvia (najmä ťažobný priemysel, výroba papiera, energetika, chemický a farmaceutický priemysel, ropný a plynárenský priemysel, námornú dopravu...) Patrí sem aj CZOPC sídliace v Ostrave. Na fotke môžete vidieť nové sídlo divízie procesnej automatizácie.

2.2 Popis pracovného zaradenia študenta

Pri pohovore do tejto firmy padli otázky mne blízkych technológií, ktoré by som vedel vo firme používať a na základe ktorých ma mali priradiť ku konkrétnemu projektu. Firma pracovala na rôznych projektoch, ktoré používali najmä C#, Javu, .NET technológie, MVC, MS SQL databázy, javascript, jquery a pod. Mne pri náboře najviac vyhovovali C#, .NET a MS SQL, na základe ktorých som bol priradený k príslušnému projektu, ktorý používal tieto konkrétne technológie. Keďže väčšina projektov sa vyvíjala pre zahraničných zákazníkov požadovali tiež pokročilú znalosť angličtiny, ktorú som mal. Pracoval som na pozícii softvérového vývojára pre divíziu procesnej automatizácie. Táto divízia patrila pod CZOPC v Ostrave, kde sa nachádzali softvéroví vývojári, aplikační inžinieri a pod.

3 Použité technológie a prístupy

Pri práci na projekte som používal technológie .NET framework 4.5, programovací jazyk C#, javascript, s ktorým som v tejto dobe ešte nemal žiadne skúsenosti a grafické rozhranie Windows Forms. Ako programovacie prostredie sme používali Visual Studio 2012. Pri programovaní sme sa riadili metódou agilného programovania – Scrumom. V úvodných dňoch, keďže ma priradili do už rozpracovaného projektu bolo mojou primárnou úlohou príprava prostredia, nastavenie prístupov, napr. k serveru nigashi, ktoré sme používali pre tímové online zdieľanie projektu a zoznámenie sa s projektom samotným.

3.1 .NET

.NET je súbor technológií v softvérových produktoch, ktoré tvoria celú platformu, ktorá je dostupná nie len pre web a Windows. Common Language Infrastructure je štandardizovaná špecifikácia jadra .NET, ktorá rieši napríklad problém kompatibility rôznych programovacích jazykov. Základným komponentom je Microsoft .NET Framework, prostredie potrebné pre beh aplikácií a ponúkajúce ako spúšťačie rozhranie, tak aj potrebné knižnice. Pre vývoj .NET aplikácií Microsoft vydal Visual Studio. Platforma .NET nepredpisuje použitie žiadneho programovacieho jazyka. Bez ohľadu na to, v čom bola aplikácia pôvodne napísaná, sa vždy preloží do medzijazyka Common Intermediate Language. Najpoužívanejšie programovacie jazyky pre vývoj .NET aplikácií sú C#, Visual Basic a C++. Vo firme som používal najmä ASP.NET a C#.

ASP.NET – je jednotný model vývoja webových aplikácií, zahrňujúci služby nevyhnutné k zostaveniu rozsiahlych webových aplikácií s minimálnym množstvom kódu. Táto technológia je súčasťou .NET Frameworku a vďaka tomu máme pri vytváraní aplikácií prístup k triedam rozhrania .NET Framework.

C# - tento jazyk bol vyvinutý práve pre .NET Framework. Jeho syntax je prispôbená podľa dvoch najzastúpenejších a najpoužívanejších jazykov, klasické C-čko a Java a takisto je jednoduchá a zmyselná.

3.2 JavaScript

JavaScript je skriptovací programovací jazyk, ktorý sa používa hlavne pri tvorbe interaktívnych webových stránok a zapisuje sa priamo do HTML kódu. Beží na klientskom počítači návštevníka a takisto nevyžaduje neustále sťahovanie dát z navštevovaných webových stránok. (Ich protikladom sú serverové skripty, ktoré sa vykonávajú na serveri a na klienta idú už len výsledky.)

3.3 Scrum

Scrum je metóda agilného programovania, ktorá dovoľuje menším tímom v krátkych iteráciách vyvíjať produkty s dobre odhadnuteľnou cenou a podľa zákazníkových požiadaviek. Tvrdí, že znalosti vznikajú zo skúseností a rozhodnutia sa tvoria na základe toho, čo už poznáme. Scrum využíva iteračný prístup k optimalizácii predvídateľnosti. Scrum

tím sa skladá z vlastníka produktu, vývojového tímu a Scrum Mastra.

Náš tím bol spočiatku zložený z 5 ľudí, kde 1 bol vlastník produktu a zvyšní 4 sme priamo pracovali na vývoji produktu, kde 1 z nás bol scrum master. Projekt sme vyvíjali v jednotlivých šprintoch (iteráciach). Šprint, ktorý bol rozdelený do jednotlivých user story sme mali spočiatku určený na dobu 2 týždňov a počas neho sme mali vytvoriť hotový, použiteľný a potenciálne nasaditeľný prírastok produktu. Ku koncu každého šprintu sme si vyčlenili nejakú dobu na testovanie a opravovanie chýb, pričom zavŕšením každého šprintu bolo predvádzacie demo pred našim zákazníkom, ktorý si vždy určil, či chce vo vývoji pokračovať a ak áno, čo chce mať do ďalšieho predvádzacieho dema hotové. Počas vývoja na projekte sa náš tím trhal s príchodom nových projektov a na konci sme ostali vyvíjať tento projekt už len 2 a trvanie iterácie bolo mesiac.

4 Zoznam úloh zadáných študentovi počas odbornej praxi

Firma ma priradila k už rozpracovanému webovému projektu „Monthly record application“. Tento projekt sa v súčasnosti zaoberá hlavne odpracovanými hodinami vo firme strávenými jednotlivými zamestnancami na rôznych projektoch s rôznymi činnosťami na nich a pod. Úlohou bolo vytvoriť viacero prepočtov s hodinami obsahujúcich množstvo výnimiek a špeciálnych prípadov, tieto a rovnako aj všetky potrebné informácie čo najprehľadnejšie zobrazíť na príslušnej stránke a umožniť editácie a úschovu množstva údajov za behu s neovplyvnením predošlých a umožnením užívateľom pozeráť svoju históriu. Výstupom tejto aplikácie sú mimo iného aj rôzne typy excelov a pdf súborov s detailným popisom obsahujúcim všetky potrebné údaje o projektoch a zamestnancoch.

4.1 Projekt “Monthly record application”

Aplikácia Monthly record application sa primárne zaoberá odpracovanou dobou zamestnancov vo firme. Zahrňuje údaje o zamestnancoch ako sú ich osobné údaje, popisy ich mesačných sadzieb, ktoré sa môžu líšiť v čase a projekty na ktorých pracujú. Tieto si užívatelia vyplňujú na najpoužívanejšej stránke „Monthly record“. Existujú tu 2 typy projektov (P1 a P2). P2 projekty sú projekty, ktoré obsahujú činnosti ako napr. dovolenka, náhrada cez časov, návšteva doktora a pod. Podľa predvolených nastavení majú všetci zamestnanci povolené písať si hodiny na tieto projekty. Aby si zamestnanec mohol vyplňovať projekty typu P1, musí mu byť tento projekt priradený niekým (s vyššou rolou) z vedenia firmy. Tento projekt má zamestnanec priradený na určité obdobie. Pre každý projekt je nutné priradiť jednotlivé objednávky a ich definície. Po priradení si môže zamestnanec z ponuky vybrať pre konkrétny projekt prislúchajúce objednávky, kde pre každú konkrétnu objednávku prislúchajú priradené definície tejto objednávky. Každá kombinácia týchto 3 výberov predstavuje 1 riadok v našej aplikácii. Pre tento riadok si takisto môže vybrať z preddefinovaných aktivít, ktoré popisujú aktivitu strávenú na projekte. Zamestnanec môže v 1 deň robiť na viacerých projektoch, objednávkach, alebo definíciách a pre každý výber si vyplňuje hodiny a píše komentár, čo v daný deň pre vložený počet hodín robil. Zamestnanci si môžu takisto písať hodiny na Home Office, kde im musí byť práca na Home Office povolená. Aplikácia počíta odpracované hodiny na projektoch rôzne v závislosti z vybranej ponuky možností a takisto počíta cez časy, ktoré sa počítajú na základe určitých špecifických pravidiel. Aplikácia obsahuje niekoľko stránok, kde každá z týchto stránok sa zobrazuje vybraným užívateľom na základe role im priradenej. V aplikácii rozlišujeme 7 typov rolí, kde medzi najnižšie role patria zákazníci a používatelia a najvyššiu rolu má administrátor. Špecifickými stránkami v našej aplikácii sú okrem hlavnej stránky „Monthly record“ stránka pre schvaľovací proces, stránka s nastavením jednotlivých užívateľov, ďalej stránka, kde sa definujú projekty, objednávky a definície, stránka s globálnymi nastaveniami v aplikácii ako sú sviatky, krajiny a podobne a za zmienku stojí takisto stránka obsahujúca výstupné reporty, ktorých nie je málo a delia sa podľa definovaných kritérií. Aplikácia pred mojim nástupom do firmy obsahovala tieto stránky, kde každá z nich mala jednoduchý dizajn. Obsahovala základné členenia s minimálnym počtom nevyhnutných atribútov, možností a neobsahovala skoro žiadne reporty. Komplexnosť a dizajn tejto ap-

likácie sa za moje pôsobenie vo firme výrazne zmenil a pribudlo tu nemálo kritérií a špecifických prípadov, kde sa každé z nich museli riešiť iným spôsobom.

4.2 Vykonané úpravy, rozšírenia a ich stručný popis

V tejto aplikácii som vykonal nemálo úprav a rozšírení a taktiež som sa tu stretol s nejedným problémom, pričom tu opíšem iba niekoľko vybraných, na ktorých som v tejto firme pracoval. Jednou z prvých úloh, ktoré mi tu boli priradené, bola práca s aktivitami na priradených projektoch, kde pribudli nové aktivity a ich definície a tiež kalkulácia hodín v závislosti od konkrétnej aktivity. Riešil som tu najmä problémy spojené s kalkuláciou a prepočtami hodín, ktoré sa ďalej prejavili vo výstupných reportoch. Jednou z ďalších úloh boli úprava dizajnu hlavnej tabuľky pre vkladanie hodín na prislúchajúce projekty a jej dynamické posúvanie v závislosti od rozlíšenia obrazovky, označovanie konkrétnych hodnôt po kliknutí myškou, povolenie a blokovanie špeciálnych klávesníc, kde po kliknutí na niektoré sa spustili nechcené veci, alebo naopak, po stlačení vybranej klávesy sa požadované veci nespustili. Neskôr som riešil úlohu, kde bolo požadované posielanie informačných mailov s nadefinovaným textom, ktorý mohol obsahovať prídavný text s informačnou správou od vybraných ľudí s dostatočnými právami, ktorý tieto maily mohli posilať a ktoré posielali zvoleným užívateľom. Pokračoval som úlohou, ktorá obsahovala centrovanie popupov. Neskôr som mal urýchliť pracovanie s datami uloženými v databázy, kde bolo pre prepis mapovania pre prístup k dátam určené PetaPoco (pôvodne sme používali Linq). Na základe špecifikácii som mal umožniť aby daný projekt začínal, alebo končil uprostred mesiaca a keďže stránka pre vyplňovanie hodín na projektoch je rozdelená na základe mesiaca, tak som musel zablokovať vkladanie hodnôt pre dané dni v tomto mesiaci a pre iné projekty v tomto mesiaci, ktoré to mali umožňovať to zase povoliť. Nutné bolo v aplikácii umožniť refresh stránky v špecifických situáciách. Ďalšou úlohou bola zmena štruktúry projektov, kde pribudla nová úroveň stromovej štruktúry a tiež prepísanie treeview do dropdownlistov. Medzi jednu z tých zaujímavejších úloh patrilo posúvanie väčšiny popupov spoločne so scroll barom a zašednutie pozadia za popupmi, kde vznikol najmä problém s Internet Explorerom 8, pre ktorý sme vyvíjali a kde všetko nefungovalo ako v novších verziách, alebo konkurenčných prehliadačoch. Ďalšia úloha, ktorú tu opíšem je práca s exportom údajov do excelu, kde som musel použiť nový spôsob exportu, lebo data mali byť rozdelené do excelovských worksheetov a toto nám doterajší spôsob exportu neumožňoval. Na záver tu opíšem problém spojený so špeciálnymi znakmi, ako je napríklad apostrof, ktorý nám zhodil aplikáciu nasadenú na produkčnom serveri.

4.3 Práca s aktivitami na priradených projektoch, kalkulácie hodín

Mojou prvou úlohou vo firme bolo vytvoriť typy pre príslušné aktivity. Doteraz boli všetky aktivity označené ako bežné aktivity a na mne bolo vytvoriť rozlišovanie medzi aktivitami, a to či je zvolená aktivita A, aktivita B, alebo aktivita C. Ďalšou časťou tejto úlohy bolo zamedzenie vyplneniu a vypísanie varovnej správy po vyplnení viac ako špecifikovaný počet hodín pre každý deň a pre všetky riadky, ktoré boli označené aktivitou

B. Pre splnenie tejto funkcionality bola nevyhnutná práca s javascriptom, s ktorým som sa dovtedy nestretol. Mojimi ďalšími úlohami bolo pokračovanie v práci s počítaním hodín a ich rozlišovaním podľa zvolenej aktivity a farebnému rozlišovaniu vyplnených hodín, kde bola akceptovaná špecifikovaná vyplnená suma hodín v 1 dni pre daný typ projektov, pre víkendy a sviatky bola určená iná suma, pre projekty označené aktivitou B bol taktiež určený iný rozsah hodín a tiež pre aktivitu C pre tento deň. Ku koncu iterácie sme pri testovaní odhalili chybu, kde sa do celkových súm zle počítali hodiny na novo pridanom riadku, alebo zmazanom už vyplnenom riadku a tak mi bolo priradené nájsť a opraviť túto chybu. Takisto som mal zablokovat stlačenie tlačidla Enter pre bunky, kde sa vyplňovali odpracované hodiny. Starý spôsob počítania súm hodín pre jednotlivé riadky a stĺpce na základe aktivít bol už tak komplikovaný a obsahoval už príliš moc vnorených podmienok a vetvení, že sa muselo prepísať počítanie z pôvodného počítania na základe riadkov na počítanie na základe stĺpcov podľa týchto aktivít. Následne mi bolo zverené upraviť počítanie cez časov a ak mal užívateľ na aktivite B napísaných viac ako je povolený počet hodín, mali sa tieto hodiny pripísať do cez časov. Nasledovala úloha zmeniť prácu s počítaním hodín, kde sa doteraz hodiny počítali ako hodiny s desatinnými hodnotami a spraviť validácie pre vkladanie iba celých čísel, spolu s výpisom varovných správ a prepísať prácu s desatinnými číslami na prácu iba s číslami celými + validácie v celej aplikácii. Špecifikácie s počítaním hodín sa neustále menili, až dospeli do fázy, kde som musel upraviť počítanie tak, aby sa hodiny priradené k aktivite C počítali do sumy pre príslušný riadok, nepočítali sa do sumy stĺpcov pre daný deň a takisto sa nepočítali do celkovej sumy. Vznikli tu výnimky, pri ktorých sa hodiny mali počítat do sumy pre daný deň, do sumy riadkov a celkovej sumy, ale nemali sa počítat do súhrnnej tabuľky, v ktorej sa zobrazovali štatistiky sumy odpracovaných hodín, cez časov a podobne. Aplikácia obsahovala aj niekoľko výnimiek pre počítanie hodín spolu v kombinácii podľa druhov projektu, ktoré sa začali počítat ináč a tiež sa tu museli zahrnúť výnimky pre hodiny, ktoré sa odpočítavali a iné.

4.4 Úprava vzhľadu a šírky hlavnej tabuľky pre projekty

Po práci s javascriptom a C# nasledovala práca s CSS. Keďže sa do hlavnej tabuľky s vyplňovaním hodín na projektoch neustále pridávali nové veci a zväčšovala sa šírka dropdownlistov, tak sa už na menšiu obrazovku nezmestili všetky dôležité informácie a muselo sa posúvať lištou, aby bolo vidieť všetko potrebné. Mojou úlohou bolo v tomto prípade upraviť CSS, aby bolo aj na menšej obrazovke vidieť všetky potrebné informácie. Upravoval som tu hlavne veľkosť celej tabuľky, ktorá bola zložená z dropdownlistov a buniek, v ktorých boli obsiahnuté tlačidlá, po kliknutí na ktoré sa zobrazil popup, textboxy pre vkladanie odpracovaných hodín a ešte nejaké štítky.

4.5 Povolenie a blokovanie špeciálnych kláves

Podľa požiadavkou užívateľov sme museli v našej aplikácii povoliť priechod medzi textboxami pre vkladanie hodnôt stláčaním klávesnice Tab, ktoré doteraz nefungovalo. Takisto

som musel označiť vložené hodnoty v textboxoch, na ktorých bol focus pre zjednodušenie editácie tejto hodnoty. Ak bolo nejaké z tlačidiel, ktoré otváralo popupy označené, tak po stlačení klávesnice Enter sa otvoril popup pre vloženie týchto komentárov. Tomuto predvolenému nastaveniu som musel zamedziť. Riešil som to cez javascript s použitím keydown funkcie. Popup som zobrazoval iba po kliknutí na tlačidlo.

```
$(document).keydown(function (e) {
    // On press Enter
    if (e.keyCode == 13) {
        popupVisibility (false);
        ...
    }
}

...

$('#buttonId'). click (function () {
    popupVisibility (true);
    ...
})
```

Výpis 1: Blokovanie klávesnice Enter a zobrazovanie popupu

4.6 Označovanie hodnôt a posielanie informačných emailov

Po zvolení vybraných užívateľov buďto postupným zakliknutím, alebo hromadným označením všetkých užívateľov sa mal týmto užívateľom poslať informačný mail, ktorý obsahoval informácie o stave nevyplnenia vstupných hodnôt a informoval týchto užívateľov, aby si hodiny vyplnili. Mojou úlohou bolo napojiť už existujúci interný modul, pre posielanie mailov s našou aplikáciou, ktorý som pripojil cez webovú službu. Tento email mal obsahovať informácie o užívateľovi, dátume, v ktorom si nevyplnil hodiny + informačná správa s textom od človeka, ktorý tento email posielal. Email mal takisto obsahovať odkaz, ktorý sme museli vytvoriť, na ktorý keď užívateľ klikol došlo k presmerovaniu na príslušnú stránku s nevyplnenými hodinami. Vznikali tu problémy najmä s odkazom, ktorý nám miestami nefungoval, keďže bol iný v aplikácii spustenej lokálne, spustenej na testovacom serveri, alebo na nasadenom serveri. Nakoniec som upravil generovanie tohto linku na základe konfiguračného súboru. Musel som tu tiež vyriešiť zablokovanie posielania emailu po opakovanom rýchlom stlačení (niekoľkokrát za sekundu) tlačidla pre posielania emailov, ktoré predbehlo vyskakovacie okno s informáciou o poslaní emailu, ktoré musel posielajúci ešte potvrdiť.

4.7 Centrovanie vyskakovacích okien

Ďalšia úloha bola pre mňa celkom zaujímavá. 90% vkladania nových údajov, ako sú napr. vkladanie komentárov k odpracovaným hodinám, vytváranie nových užívateľov, projektov, objednávok a pod. máme riešených vyskakovacími oknami. Doteraz sa tieto okna zobrazovali v určitej oblasti na obrazovke a keď stránka obsahovala veľa údajov, tak sa

okno vedelo zobrazíť tak, že ho na obrazovke ani nebolo vidieť a muselo sa posúvať bočnou lištou, aby sa našlo a tiež aby sme videli čo doňho píšeme. Za úlohu som dostal všetky vyskakovacie okná, ktoré sa nachádzali v našej aplikácii vycentrovať na stred obrazovky, bez ohľadu či bolo okno prehliadača maximalizované, alebo zmenšené. Niekedy sa pri zobrazovaní týchto okien vytváral aj scrollbar a mojou úlohou bolo zamedziť jeho zbytočnému vytváraniu.

4.8 PetaPoco

PetaPoco je stručný, rýchly, jednosúborový micro-ORM pre .NET a Mono.

- je to 1 súbor, ktorý sa dá jednoducho pridať do projektu
- je silne typový
- podporuje tiež dynamické rozšírenie
- podporuje úzke vzťahy medzi objektami a databázovými tabuľkami
- podporuje tvorbu Poco tried spolu s T4 templejtami
- je rýchly, pretože používa dynamickú metódu vytvárania (MSIL) pre priradenie stĺpcových hodnôt vlastnostiam

Je dostupný z NuGet, alebo GitHub.

4.8.1 Prepísanie DAO Mapperu z Linq do Petapoca

Tento projekt, ku ktorému som bol priradený používal ako mapovanie s databázou Linq. Ja osobne som pred nástupom do firmy s Linqom nič nerobil a rovnaké to bolo aj s Petapocom, ktorý bol pre všetkých vo firme nový. Časom trval už prístup k dátám nepohodlne dlho a tak bolo nutné prepísať celé mapovanie na Petapoco, ktoré bolo zreteľne rýchlejšie a neobsahovalo toľko prebytočných dát, ako Linq. Pôvodný DAO mapper v našej aplikácii bol cca na 5000 riadkov a po prepísaní na Petapoco sa tento počet znížil o cca 1000 riadkov. Prístup k databáze bol znateľne rýchlejší a trval podstatne kratší čas, ale prepísanie všetkých metód nám vo dvojici zabralo asi týždeň. Pri prepisovaní metód sme niektoré staré funkcie hneď aj zrefaktorovali.

4.9 Projekt mohol začať, alebo skončiť hocikedy v priebehu mesiaca

Na vyplňovanie hodnôt sme používali rovnaké bunky pre všetky typy projektov. Ak mal užívateľ priradený iba 1 projekt, ktorý začínal neskôr, ako prvý deň v mesiaci, alebo končil skôr, ako je posledný deň v mesiaci bolo všetko v poriadku a dané bunky pre vyplňovanie hodnôt v danom dni v tomto mesiaci, ktoré boli pred začiatkom, alebo po skončení sa vytvorili bez textboxov, do ktorých by užívateľ mohol vkladať hodnoty. Tento problém nenastal, ani keď mal užívateľ priradených viac projektov, z ktorých si mohol vyberať cez dropdownlisty ten, ktorý chce vyplňovať. Komplikácie, ktoré sme museli riešiť nastali

iba v prípade, keď mal užívateľ priradenú kombináciu týchto projektov, kde aspoň 1 z nich začínal v prvom dni zvoleného mesiaca (resp. končil posledný deň v mesiaci) a aspoň 1 ďalší začínal neskôr ako prvý deň v mesiaci (resp. končil skôr ako posledný deň). Užívateľovi sa vždy z databázy primárne načítal prvý projekt z dropdownlistu, alebo ak mal na nejakom už uložené hodnoty, tak tento projekt. Keď si chcel zmeniť projekt, na ktorý by si písal hodiny z projektu ktorý napr. začínal 6.3. na projekt, ktorý začínal 1.3. a začal vyplňovať hodnoty do 1 z týchto buniek, tak aplikácia spadla, lebo aj keď sa tam textboxy po prepočítaní v javascripte vytvorili, aplikácia ich nepoznala. Riešením teda bolo zmazať prvú metódu, ktorá vytvárala bunky bez textboxov, ak projekt nemal začiatok a koniec v plnom rozsahu dní mesiaca a vytvárať tieto textboxy vždy. Na základe vybraného projektu som zisťoval dátum priradený projektu a tiež dni, ktoré neboli určené pre vyplnenie a tie som vždy cez javascript nastavil ako disabled a zmenil pozadie z bielej na snehovú farbu a ak v týchto zakázaných dňoch už boli nejaké vložené hodnoty, tak som ich zmazal a takisto som prepočítal hodnotu všetkých súm. Po tom, ako si užívateľ prepol projekt naspäť na taký, kde malo byť umožnené vkladať hodnoty pre všetky dni v mesiaci som túto voľbu povolil a zmenil pozadie naspäť na bielu. Hneď ako som to spravil, začali sme túto funkcionálnosť testovať na mojom počítači a všetko fungovalo ako malo. Asi 2 dni po tom, keď môj kolega toto testoval na jeho počítači sa vždy zmenila iba farba pozadia, ale samotné bunky sa nezablokovali a umožňovali vkladanie hodnôt do nich s tým, že sa nepripočítavali do výsledných súm. Po hlbšom skúmaní kde je chyba som prišiel na to, že tento problém bude spôsobený verziou prehliadača. Ja som mal v počítači nainštalovaný IE 10 a kolega IE9. Aplikácia, ktorú sme vyvíjali pre firmu bola určená na IE8 a vždy pri spustení aplikácie sme cez konfiguračný súbor nastavovali štandardy na oficiálne štandardy IE8, ktoré ako sa ukázalo sa chovalo ináč, aj keď sme mali nastavenú úpravu štandardov na túto rovnakú verziu (IE8). Nakoniec som tento problém vyriešil prepísaním funkcií z javascriptu do jquery, ktorý si potrebné chýbajúce veci doimportoval a následne všetko fungovalo ako malo.

```
$("#hours" + row + "Cell" + day).attr("disabled", true);
hoursCell.style.backgroundColor = "#E8E8E8"; // gray
```

Výpis 2: Blokovanie buniek pre vkladanie hodnôt

4.10 Dynamické posúvanie tabuľky s projektami

Pre veľkosť a množstvo údajov, ktoré sa mali do tejto tabuľky zmestiť, sme túto tabuľku posunuli do mínusových hodnôt o 150px, ktorá sa na monitore s dostatočne veľkým rozlíšením zobrazovala tak, že cca začiatok aj koniec vyčnievali na obidve strany o 150px a tabuľka bola vycentrovaná pekne v strede monitora. Problém, ktorý nastal v tomto prípade vznikol na monitoroch s malým rozlíšením, alebo tiež po zmenšení okna prehliadača. V tomto prípade sa hodnoty, ktoré boli posunuté na ľavej strane do mínusových hodnôt nezobrazovali a stratili sa. Hodnoty ktoré sa zobrazovali na pravej strane bolo vidieť normálne po posunutí posuvnej lišty v prehliadači. Aby ostala tabuľka vycentrovaná na stred na veľkom monitore, kde si ju väčšina užívateľov zobrazovala a zároveň bolo vidieť všetky hodnoty po zmenšení okna som musel dynamicky upravovať umiestnenie

tejto tabuľky na základe aktuálneho rozlíšenia a posúvala sa vždy okamžite so zmenou veľkosti okna. Riešil som to cez javascript a jquery.

```
function ResizeWindow() {
    var popupWindow = document.getElementById("popupWindow");
    var table = document.getElementById('table');

    if (popupWindow == null) {
        var windowWidth = $(window).width(); // returns width of browser viewport

        if (table != null) {
            if (windowWidth < 1260) {
                var numberOfPixels = (1280 - windowWidth) / 2;
                if (numberOfPixels < 150) {
                    table.style.marginLeft = numberOfPixels + "px";
                }
                else {
                    table.style.marginLeft = "150px";
                }
            }
            else {
                table.style.marginLeft = "0px";
            }
        }
    }
}
```

Výpis 3: Dynamické posúvanie tabuľky

4.11 Refresh stránky

Väčšina stránok je riešená cez update panely. V prípade, že si užívateľ zmenil na stránke s vyplňovaním údajov na odpracovaných projektoch z dropdownlistu prázdnu hodnotu, ktorú nemal vyplňovať a rovnako zmenil hodnoty v ďalších dropdownlistoch za prázdne a klikol na tlačidlo, ktoré ho malo presmerovať na rovnakú stránku, tak aplikácia spadla. Údaje uložené na tejto stránke sa niekedy zobrazovali priamo na stránke a niekedy sme ich museli zobraziť v popupe. Aplikácia padala iba po presmerovaní na rovnakú stránku, ak nebola zobrazená v popupe. Tento problém som vyriešil refreshom stránky iba ak sa údaje z tejto stránky nenachádzali v popupe, ale priamo na tejto stránke. Refresh stránky som spustil v javascripte kódom:

```
document.location.reload(true);
```

Výpis 4: Refresh stránky

4.12 Prepísanie treeview do dropdownlistov a rozšírenie stromovej štruktúry

Podľa nových špecifikácií sme museli zväčšiť stromovú štruktúru z 2 úrovní na 3. Aby bola stránka prehľadná, musel som zrušiť treeview a pridať k už existujúcemu dropdownlistu

ďalšie 2. Prvý dropdownlist naplníme rokmi, na ktorých sú uložené nejaké údaje a defaultne sa vyselektuje aktuálny rok. Na stránke sa po novom v gridview zobrazujú údaje príslušné k tomuto roku (1. úroveň stromu) z ktorých si je možné vybrať buďto priamo po zvolení nejakého riadku v gridview, alebo po vybraní cez dropdownlist. Po vybraní prvku sa zmení gridview obsahujúce prvky 1. úrovne na gridview obsahujúci prvky 2. úrovne a v 2. dropdownliste sa označí zvolený prvok. Po zvolení potomka či už cez gridview, alebo dropdownlist sa sa gridview znovu zmení na gridview obsahujúci prvky 3. úrovne, pričom sa zvolený prvok musí znova označiť v dropdownliste. Návrat na predchádzajúci dropdownlist som riešil pomocou nového tlačidla „Back“ hneď vedľa dropdownlistov. Po stlačení tlačítka som musel vždy vyprázdniť posledný dropdownlist, keďže prvky, ktoré obsahoval už nepatrili k ničomu, pretože sa presunul vybraný prvok o úroveň vyššie. Všetky gridview máme umiestnené v update paneloch, aby sme predišli zbytočnému refreshu stránky a vždy po pridaní nového prvku do dropdownlistu som si uložil všetky prvky obsiahnuté v dropdownne do listu ListItem-ov. Následne som mu priradil nový prvok a zmazal všetky prvky z dropdownu. Pokračoval som súčasným zoradením prvkov v tomto liste, ktorý obsahoval už aj nový prvok a týmito zoradenými prvkami som opätovne naplňoval dropdownlist. Podobný postup som používal aj pri editácii prvkov. Jedným z ďalších problémov, ktoré som tu musel riešiť bolo, keď sme zmenili pageIndex na niektorom z gridview, kde sme pri zmene tohto indexu museli zaregistrovať metódu PageIndexChanging a príslušnému gridview priradiť nový index takto:

```
gridview.PageIndex=e.NewPageIndex;
gridview.DataBind();
```

Výpis 5: Zmena indexu v gridview

Jeden z posledných problémov, ktorý som tu musel riešiť bolo nutnosť vyhľadávania podľa zadanej hodnoty do textboxu medzi hodnotami v aktuálne zobrazenom gridview a následne na stránke zobraziť iba tie prvky v gridview, ktoré obsahovali hľadaný reťazec. Podľa počtu prvkov a následne podľa vyselektovaného prvku v dropdownliste som pri priechode od dropdownlistu najnižšej úrovni zistil aktuálnu úroveň, kde sa užívateľ nachádza a podľa toho som naplňoval gridview hodnotami, ktoré obsahovali hľadaný prvok. Takisto som tu pridal funkcionality hľadania po stlačení klávesy Enter.

```
List<ListItem> oldList = new List<ListItem>();

foreach (ListItem item in dropDownList.Items)
{
    if (item.Value != string.Empty && int.Parse(item.Value) == id)
    {
        item.Text = text;
        item.Value = value;
    }
    oldList.Add(item);
}
dropDownList.Items.Clear();

foreach (ListItem item in oldList.OrderBy(itemValue => itemValue.Text))
```

```
{  
    dropDownList.Items.Add(item);  
}
```

Výpis 6: Vkladanie hodnôt do dropdownlistov a zoradenie

4.13 Posúvanie popupov spoločne so scroll barom a zošednutie obrazovky za popupmi

Na túto úlohu som sa zo začiatku tešil, lebo pre mňa predstavovala novú výzvu a neskôr mi priniesla nové vrásky na tvári, pre problémy spojené s IE. Doteraz sa v našom projekte všetky vyskakovacie okná zobrazovali na strede obrazovky s tým, že prekryli údaje, ktoré boli zobrazené na stránke za nimi iba veľkosťou okna a nebola tu zablokovávaná možnosť užívateľa kliknúť na nejaký prvok (napr. tlačidlo), ktorý sa vyskytoval na stránke za týmto popupom a spustiť tým novú funkcionality. Rozhodol som sa tento problém vyriešiť zašednutím všetkého, čo bolo zobrazené na stránke a bolo vidieť okolo tohto vyskakovacieho okna. Problém som skúšal riešiť niekoľkými rôznymi spôsobmi, kde najefektívnejšou sa zdalo využitie vrstiev. Mojou snahou bolo rozdeliť to čo sa zobrazovalo na stránke od toho, čo sa zobrazilo v popupe a vložiť medzi ne novú medzivrstvu, ktorá by umožňovala vidieť to čo je na stránke okolo popupu, ale neumožňovala na prvky, ktoré sa tam nachádzali klikať, alebo iným spôsobom k nim pristupovať. Po pridaní novej medzivrstvy a následnému zobrazeniu tejto vrstvy sa táto vrstva nezobrazila. Neskôr som zistil, že je to spôsobené tým, že sme mali všetky prvky zobrazené na stránke v nejakom dive, ktorý obsahoval ďalšie divy a aby bola táto medzivrstva zobrazená, tak by nemala byť predtým obsiahnutá v hlavnom dive tejto stránky, lebo vždy dedí vlastnosti od nadradeného prvku a následne nejdú prepísať. Musel som teda prvky, ktoré sa vyskytovali na stránke porozdeľovať do viacerých menších divov, kde som vytvoril aj tento nový, ktorý bol určený pre zošednutie a dedil vlastnosti z CSS triedy. Takisto som všetkým divom na stránke musel upraviť vlastnosť z-index, ktorú sme doteraz mali v aplikácii nastavenú na rôzne nezmyselné hodnoty s tým, že sa doteraz táto vlastnosť ani nevyužívala. Po kompletnom prepísaní všetkých divov na stránkach, ktoré obsahovali vyskakovacie okná fungovalo všetko ako malo.

4.14 Export údajov do excelu obsahujúcich worksheetsy

Jednou z posledných zaujímavejších vecí bol export všetkých údajov obsiahnutých na stránke do excelu. Údaje mali byť porozdeľované podľa obsiahnutých prvkov do excelovských worksheetsov. V aplikácii sme mali doteraz už niekoľko exportov údajov do excelov, kde žiadne z týchto excelov doteraz neobsahovali worksheetsy. Exporty sme riešili cez vytvorenie inštancie DataTable, kde sme plnili jednotlivé stĺpce a riadky hodnotami. Nakoniec sme si vytvorili inštanciu GridView, ktorej sme predali ako DataSource túto tabuľku a gridView sme naplnili cez DataBind. Následne sme data exportovali do excelu s použitím StringWriteru a HtmlTextWriteru. Použitie tejto metódy v tomto prípade nešlo, keďže nič z tohto nepodporovalo vytvorenie excelovských worksheetsov. Zo začiatku

som toto riešil importom nového dll-ka, ktoré bolo priamo od Microsoftu a umožňovalo vytváranie týchto worksheetov. Po odskúšaní fungovalo všetko v poriadku, ale háčik bol v tom, že nereagovalo na odpoveď od užívateľa a hneď po zavolaní metódy sa excel vytvoril a aj otvoril. Nutné bolo aby stroj, z ktorého sa aplikácia spúšťala mal už excel nainštalovaný. Po nasadení skúšobnej verzie na testfield, kde sme tento excel nainštalovaný nemali sa excel nevytvoril. Nutná bola teda zmena dll-ka za iné, ktoré by vytváralo excel na strane užívateľa. Po konzultácii s ostatnými zamestnancami firmy mi odporučili používať EPPlus, ktorý už bol vo firme osvedčený a ktorým som tento export vyriešil.

```
ExcelWorksheet worksheet = null;
List<string> SheetNames = new List<string>();
MemoryStream ms = new MemoryStream();
using (ExcelPackage package = new ExcelPackage(ms))
{
    if (countOfData != 0)
    {
        package.Workbook.Worksheets.Add("NameOfWorkSheet");
    }

    int rowIndex = 2;
    object1.Object2.ForEach(object2Value =>
    {
        worksheet = package.Workbook.Worksheets[rowIndex + 1];

        using (ExcelRange range = worksheet.Cells[rowIndex, 1, rowIndex, 2])
        {
            range.Merge = true;
            range.Value = DateTime.Now.ToString("d.M.yyyy") + ", " + DateTime.Now.Hour + ":" +
                DateTime.Now.Minute;
            range.Style.Font.Bold = true;
            range.Style.Fill.PatternType = ExcelFillStyle.Solid;
            range.Style.Fill.BackgroundColor.SetColor(Color.White);
            range.Style.HorizontalAlignment = ExcelHorizontalAlignment.Center;

            range.Style.Border.Left.Style = ExcelBorderStyle.Medium;
            range.Style.Border.Left.Color.SetColor(System.Drawing.Color.Black);
            range.Style.Border.Right.Style = ExcelBorderStyle.Medium;
            range.Style.Border.Right.Color.SetColor(System.Drawing.Color.Black);
            range.Style.Border.Bottom.Style = ExcelBorderStyle.Medium;
            range.Style.Border.Bottom.Color.SetColor(System.Drawing.Color.Black);
        }
        rowIndex += 2;

        worksheet.Cells[rowIndex, 2].Value = "Value";
        worksheet.Cells[rowIndex, 2].Style.Fill.PatternType = ExcelFillStyle.Solid;
        worksheet.Cells[rowIndex, 2].Style.Fill.BackgroundColor.SetColor(Color.WhiteSmoke);

        ...
    }
}

#region Save and export
```

```

Response.Clear();

Response.ContentType = "application/vnd.openxmlformats-officedocument.spreadsheetml.sheet";
Response.AddHeader("Content-Disposition", "attachment;filename=" + someFileName + ".xlsx");

Response.BinaryWrite(package.GetAsByteArray());
Response.End();

package.Save();
#endregion
}

```

Výpis 7: Export údajov do excelu

4.15 Problém so špeciálnymi znakmi a zhodenie aplikácie na produkčnom serveri

Posledný problém, ktorý tu opíšem je spojený so špeciálnymi znakmi, ako je apostrof, dvojité úvodzovky a podobne. Jedného dňa nám na aplikácii, ktorá už bola nasadená na ostrom serveri a používaná zamestnancami firmy prestali fungovať najpoužívanejšie stránky pre vkladanie hodín a prácu s projektami. V tento deň sa blížil koniec mesiaca a na našu aplikáciu sa prihlasovalo najväčšie množstvo ľudí. Postupom času nám začali chodiť maily od týchto užívateľov, že naša aplikácia nefunguje ako má a keďže som bol v tento deň vo firme sám z ľudí, ktorí sme pracovali na tejto aplikácii bolo mojou prvou úlohou nájsť príčinu zhodenia aplikácie a čím skôr ju spojzdať. Po tom, ako som si prešiel kód spojený so stránkami, ktoré padali a nenašiel som tu žiadnu príčinu prečo by to malo padať, ostávalo tento problém hľadať už len priamo v databáze. Napadlo mi, že by mohol byť spojený vložením určitého špeciálneho znaku. Keďže sme už v tejto dobe mali v databáze veľké množstvo dát a ručné prehľadávanie by trvalo niekoľko hodín, musel som nájsť lepší spôsob, ako tieto podozrivé znaky nájsť. V databáze som si vytvoril procedúru, ktorá mi prechádzala všetky tabuľky z databázy a hľadala tam znak, ktorý som jej predal ako argument. Po tom, ako som tieto znaky našiel, skúšal som ich mazať, alebo nahradzovať s podobnými znakmi na lokálnej kópii produkčnej databázy, až kým som neprišiel na to, že tento konkrétny problém bol spojený vložením apostrofu. Keď som tento problém vyriešil a aplikácia už fungovala normálne, bolo mojou ďalšou úlohou umožniť vloženie tohto znaku bez toho, aby táto aplikácia znovu spadla. Mapovanie do databázy máme robené parametrizovanými dotazmi, takže v tomto smere chyba byť nemohla. Zistil som, že chyba nastala pri posielaní údajov zo C# do javascriptu, kde v C# tento apostrof problémy nerobil, ale keď sa poslal do javascriptu, tak predčasne ukončil reťazec a to spôsobilo spadnutie aplikácie. Problém som nakoniec vyriešil prepísaním kódovania týchto znakov pred poslaním do javascriptu.

```
HttpUtility .JavaScriptStringEncode(attributeWithSpecialCharacter);
```

Výpis 8: Posielanie špeciálnych znakov zo C# do javascriptu

5 Využitie znalosti a skúsenosti

Keďže som pred nástupom na vysokú školu nepoznal ani jeden programovací jazyk a všetky programovacie jazyky som sa učil za behu počas školy a víkendov strávenými doučovaním vecí do školy, z ktorých mali ostatní už aspoň nejaké základy mi škola dala dobrý základ. Najväčšie skúsenosti zo školy som získal pri tvorbe projektov. Využíval som najmä znalosti, pri ktorých som pracoval s databázami, C# a návrhovými vzormi. Predmety, ktoré mi boli najviac osožné boli Úvod do databázových systémov, Databázové a informačné systémy, Programovacie jazyky II, Vývoj informačných systémov, Úvod do softwarového inžinierstva a Algoritmy I, II.

5.1 Chýbajúce znalosti a skúsenosti

Zo začiatku mi chýbali najmä skúsenosti s prácou v tíme, zdieľaním projektu medzi viacerými ľuďmi a podobne. Takisto mi spočiatku dosť chýbali znalosti na prácu s javascriptom, ktoré som musel dohnať. Musel som sa tiež doučiť prácu spojenú s exportom jednotlivých údajov do pdf a excelovských súborov, s ktorými som musel pracovať pomerne často. Na záver mi chýbala prax, ktorú som pred praxou v tejto firme nemal.

6 Záver

Prax vo firme mi pomohla previesť veľa teoretických znalostí do praxe a získať tým nevyhnutné skúsenosti, ktoré budem v budúcnosti potrebovať pri práci na väčších projektoch. Naučil som sa tu prácu s novými jazykmi a najmä prácu v tíme a veci spojené s tímovou prácou. Získal som tu aj nové praktické, ale aj teoretické znalosti pri práci s novými technológiami a frameworkami, kde som ešte všetky nestihol previesť do praxe a zistil som tiež dôležitosť znalosti angličtiny pri práci v tímoch na zahraničných projektoch. Týchto 50 dní praxe mi dalo veľa, ale stále je ešte veľa vecí, ktoré by som sa chcel naučiť a na to by táto doba nestačila.

7 Literatúra

[1] ABB GROUP. The ABB Group a Automation and Power Technologies [online]. 2012.

Dostupné na: <http://www.abb.cz> a <http://www.abb.cz/cawp/czabb014/3747636fd47bf16bc12575a100>

[2] Scrum Guide™ - Úplný sprievodca Scrumom (Ken Schwaber a Jeff Sutherland).

Dostupné na: <https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide-SK.pdf#zoom=100>

[3] Petapoco (topten software) Dostupné na: <http://www.toptensoftware.com/petapoco/>

[4] .NET technológie Dostupné na: <http://msdn.microsoft.com/library>